

AMENDMENTS TO THE CLAIMS

Please amend the present application as follows:

Claims

1-25 (Canceled)

26. (New) A multifunctional peripheral device, comprising:

a memory device having a memory capacity selected to store a subset of a plurality of firmware code segments, wherein each of the plurality of firmware code segments is executable to implement a corresponding function in a plurality of functions of the multifunctional peripheral device, and further wherein the memory capacity is selected to preclude storing the plurality of firmware code segments in their entirety; and

a control circuit configured to execute a selected firmware code segment from amongst the subset of firmware code segments stored in the memory device.

27. (New) The peripheral device of claim 26, further comprising circuitry configured to communicatively couple with a host computer having stored thereon the plurality of firmware code segments from which the peripheral device obtains the subset of firmware code segments.

28. (New) The peripheral device of claim 26, wherein the subset consists of the selected firmware code segment.

29. (New) The peripheral device of claim 26, wherein the memory device is configured to store a first flag which when set is indicative of a presence of the selected firmware code segment in the memory device.

30. (New) The peripheral device of claim 29, wherein the first flag which when reset is indicative of an absence of the selected firmware code segment in the memory device.

31. (New) The peripheral device of claim 29, wherein the memory device is further configured to store a version indicator for indicating a version of the selected firmware code segment stored in the memory device.

32. (New) A method for supporting a plurality of operating modes in a peripheral device, the method comprising:
- reading a first flag stored in a volatile memory of the peripheral device;
 - upon detecting a set condition of the first flag, retrieving from the volatile memory a first firmware code segment corresponding to a first operating mode in the plurality of operating modes;
 - executing the firmware code segment; and
 - else upon detecting a reset condition of the first flag, wherein the reset condition is indicative of an absence of the first firmware code segment in the volatile memory, loading the first firmware code segment into the volatile memory.
33. (New) The method of claim 32, further comprising:
- providing a host computer; and
 - loading a plurality of firmware code segments into a memory of the host computer, wherein each of the plurality of firmware code segments is executable to implement a corresponding operating mode in the plurality of operating modes of the peripheral device.
34. (New) The method of claim 33, wherein loading the first firmware code segment into the volatile memory of the peripheral device comprises transferring the first firmware code segment from the memory of the host computer to the volatile memory of the peripheral device.
35. (New) The method of claim 34, wherein loading the first firmware code segment into the volatile memory of the peripheral device further comprises deleting a second firmware code segment that had been previously loaded into the volatile memory of peripheral device, the second firmware code segment corresponding to a second operating mode in the plurality of operating modes.
36. (New) The method of claim 34, further comprising:
- transferring data of an application program from the host processor to the peripheral device prior to reading the first flag stored in the volatile memory of the peripheral device.
37. (New) The method of claim 33, wherein loading the plurality of firmware code segments into the memory of the host computer comprises accessing a server.

38. (New) The method of claim 37, wherein the server is a web server and the host computer is communicatively coupled to the web server through the Internet.
39. (New) The method of claim 32, further comprising:
reading a version indicator stored in the volatile memory of the peripheral device, for determining a version of the first firmware code segment stored in the volatile memory of the peripheral device.
40. (New) The method of claim 39, further comprising:
executing an integrity check program for verifying integrity of the first firmware code segment stored in the volatile memory of the peripheral device.
41. (New) A computer network, comprising:
a multifunctional peripheral device containing a volatile memory having a memory capacity selected to store a subset of a plurality of firmware code segments, wherein each of the plurality of firmware code segments is executable to implement a corresponding function in a plurality of functions of the multifunctional peripheral device; and
a host processor comprising a memory in which is stored each of the plurality of firmware code segments, the host processor communicatively coupled to the multifunctional peripheral device for transferring the subset of firmware code segments to the peripheral device on an as-needed basis.
42. (New) The computer network of claim 41, wherein the memory capacity of the volatile memory precludes storing of the plurality of firmware code segments in their entirety.
43. (New) The computer network of claim 42, wherein the subset consists of a single firmware code segment.
44. (New) The computer network of claim 43, wherein the memory is configured to store a first flag which when set is indicative of a presence of the single firmware code segment in the volatile memory.

45. (New) The computer network of claim 44, further comprising a server coupled to the host processor through a communications network.
46. (New) The computer network of claim 45, wherein the host processor is configured to obtain firmware updates for the plurality of firmware code segments from the server.